

P_FUN

Table des matières

1	Analyse préliminaire	3
1.1	Introduction	3
1.2	Objectifs.....	3
1.3	Gestion de projet	3
2	Analyse / Conception.....	3
2.1	Domaine des données.....	3
2.2	Analyse Technique	4
2.2.1	Services Principaux et Modèles de Données	4
2.2.2	Gestion des Graphiques et Modèles de Vue	4
2.2.3	Composants UI et Filtrage	5
2.3	Analyse fonctionnelle.....	6
2.3.1	Filter Data by Energy Type	6
2.3.2	Filter Data by Date Range	6
2.3.3	Display Filtered Data in Graph.....	7
2.3.4	Save Filter Settings.....	7
2.3.5	Maquette de l'application	8
2.4	Stratégie de test.....	9
3	Réalisation.....	10
3.1	Points de design spécifiques	10
3.1.1	Gestion des données.....	10
3.2	Déroulement	10
3.2.1	Fonctionnalité Filtrage	10
3.2.2	Fonctionnalité Filter Preset	11
3.2.3	Technologie utilisée	11
3.3	Mise en place de l'environnement de travail.....	13
3.4	Description des tests effectués	13
3.5	Erreurs restantes	15
4	Conclusions	15
5	Annexes.....	17
5.1	Journal de travail	17

1 Analyse préliminaire

1.1 Introduction

Le projet P_Fun a été réalisé en parelle avec le module I323 pour mettre en pratique les notions de la programmation fonctionnel.

L'objectif du projet est de traiter un ensemble de données et de les afficher avec un graph, en **C#**

Pour les données, ainsi et technologies, c'est libre choix. Tant que le projet utilise les notions/syntaxe de la programmation fonctionnel et le graph affiche les données avec les deux axe (temps et valeurs).

1.2 Objectifs

Comme objectifs il faut trouver un jeu de données, ensuite savoir comment les traiter en **C#**, pour ensuite les utiliser pour gêner un graph.
Afficher un minimum de 3 courbes, et filtrer les données selon les dates.

1.3 Gestion de projet

Le projet a été réaliser de façon agile, et sa gestion a été réaliser avec l'application web **IceScrum**, tous les détails sur les fonctionnalités, user stories, tâches à faire, et les tests du projet sont dessus.

Git et **Github** ont été utilisé a comme **CVS**.

2 Analyse / Conception

2.1 Domaine des données

Le domaine du data set est sur la production et consommation de l'énergie en Suisse.

Le data set comprend plus de 30 de données, diviser par les 12 mois, les champs de ces données sont :

- **Production nationale centrales hydrauliques / nucléaires / thermiques classique et renouvelables**
- **Production Total**
- **Pompage d'accumulation**
- **Production nette**
- **Importation***
- **Exportation***
- **Consommation du pays**

- Pertes
- Consommation finale
- Solde exportateur / importateur

* dès l'année 2000

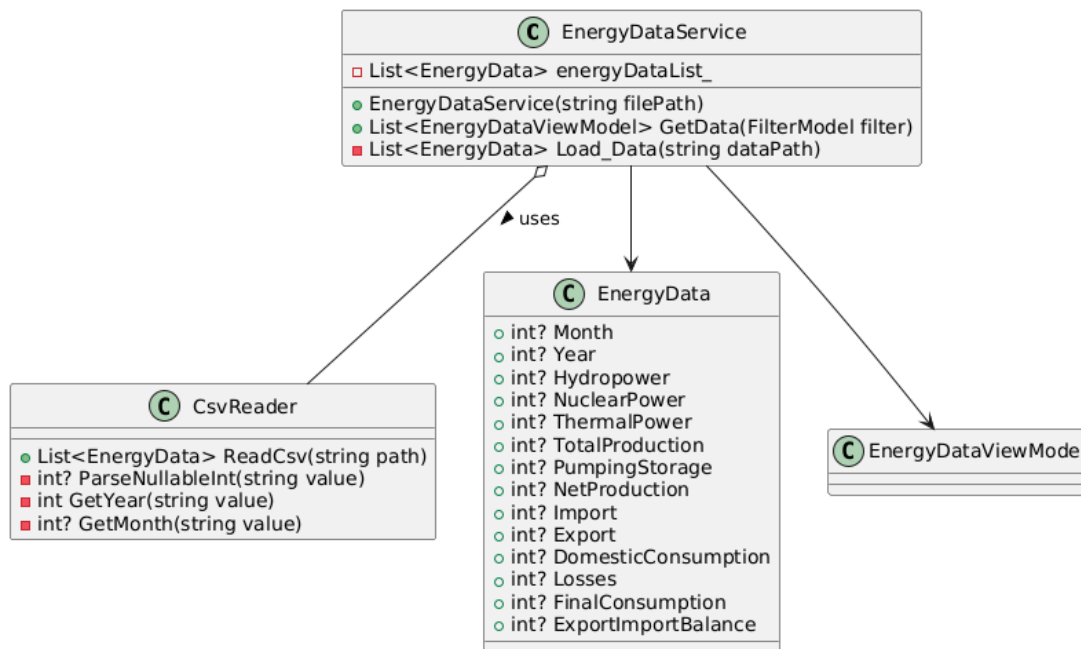
Le data set a été trouvé sur bfe.admin.ch, et a été fait par l'office fédéral de l'énergie OFEN. C'est destiné au public général, mais peut aussi être utilisé par des analystes pour :

- La balance commerciale énergétique
- L'équilibre entre production et consommation
- Saisonnalité et tendances de long terme

2.2 Analyse Technique

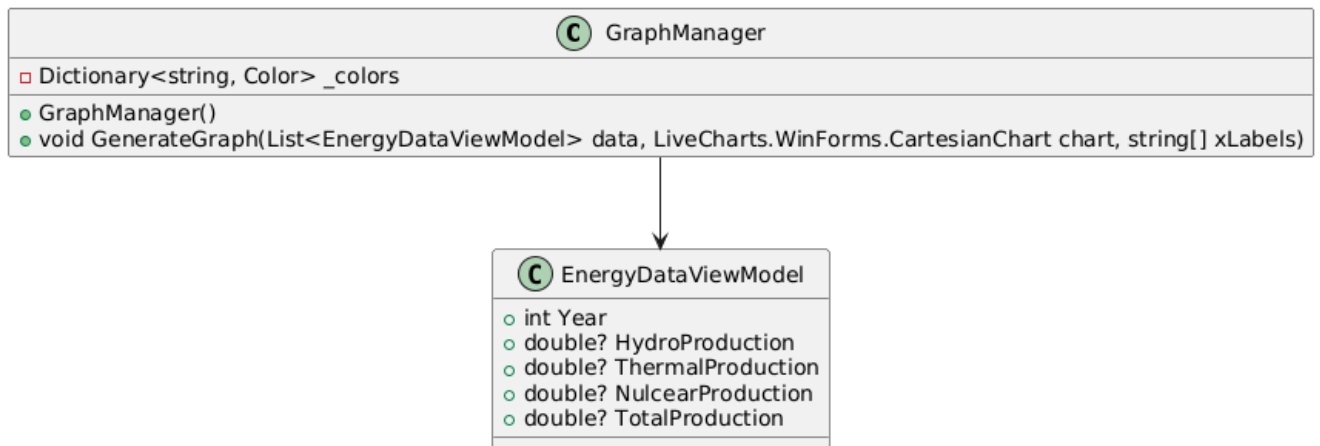
2.2.1 Services Principaux et Modèles de Données

Ce diagramme couvre les classes **EnergyDataService**, **CsvReader**, et **EnergyData**, ainsi que leurs interactions.



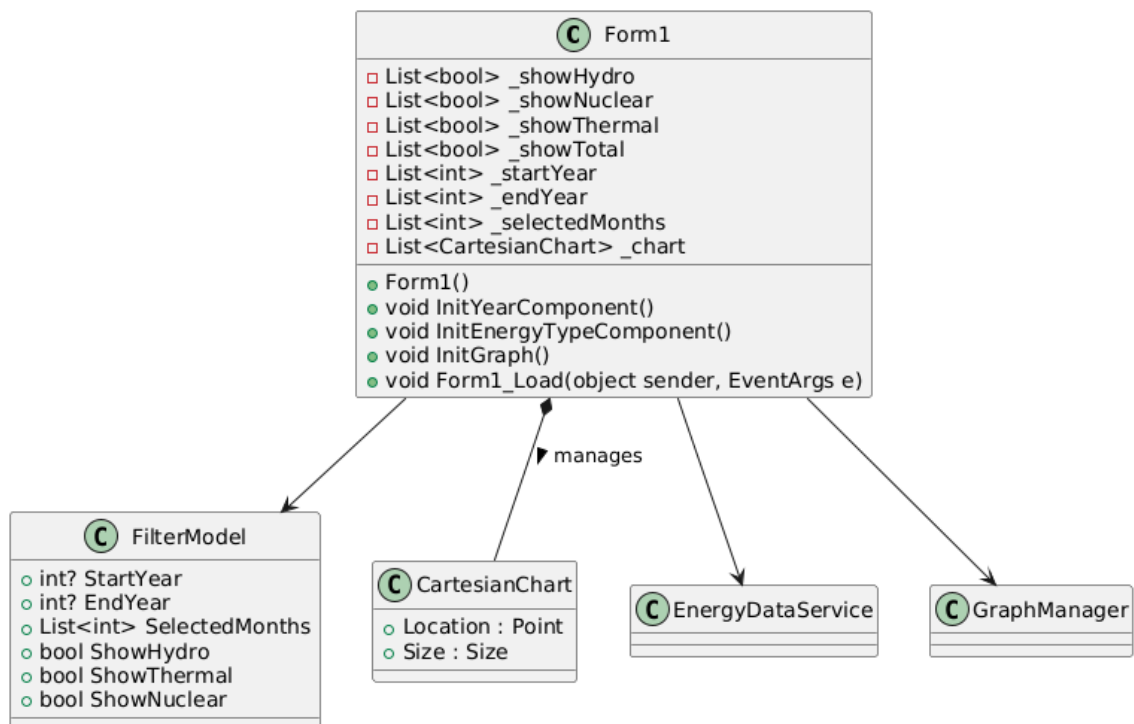
2.2.2 Gestion des Graphiques et Modèles de Vue

Ce diagramme inclut le **GraphManager**, le **EnergyDataViewModel** et leurs interactions pour la gestion des données de graphique et des séries.



2.2.3 Composants UI et Filtrage

Cette partie couvre la classe **Form1**, le **FilterModel** pour le filtrage des données, ainsi que le composant **CartesianChart** utilisé pour afficher les graphiques.



2.3 Analyse fonctionnelle

2.3.1 Filter Data by Energy Type

As a user, I want to filter energy data by type (e.g., gas, solar, wind, etc.), so that I can view only the energy types that are relevant to my analysis.

Tests d'acceptance:

**Test Filter Selection
Updates Graph**

Given that I am on the data filtering interface,
When I select one or more energy types,
Then the graph updates to display only the data for the selected types.

**Test No Filter Displays
All Energy Types**

Given no energy type is selected,
When I submit the filter,
Then the graph displays data for all energy types.

2.3.2 Filter Data by Date Range

As a user, I want to filter energy data by a specific date range (start and end date), so that I can focus my analysis on relevant time periods.

Tests d'acceptance:

**Test Date Range Filter
Updates Graph**

Given that I input a start and end date,
When I apply the filter,
Then the graph updates to show data only within the selected date range.

**Test Invalid Date Range
Triggers Error**

Given on the date filter section,
When the selected end date is earlier than the start date,
Then the filter button is unclickable.

**Test Missing Start or
End Date Triggers Error**

Given on the date filter section,
When an incomplete date is entered,
Then the filter button is unclickable.

**Test No Date Selected
Shows Full Data Range**

Given no date is selected,
When I apply the filter,

Then the graph should display data for the full time period (past 30 years).

2.3.3 Display Filtered Data in Graph

As a user, I want the filtered energy data to be displayed in a graph with clear axes, so that I can easily visualize trends over time.

Tests d'acceptance:

Test Graph Displays Correct Axes Labels

Given that I have applied filters for energy type and date range,
When the graph is generated,
Then the graph shows the correct axes for time (x-axis) and energy production (y-axis).

Test Multiple Energy Types Display with Distinct Colors

Given that I have selected multiple energy types,
When the graph is displayed,
Then each energy type should have a distinct color or label for clarity.

Test No Data for Selected Filters Shows Message

Given that no data exists for the selected filters,
When the graph is displayed,
Then I should see a message saying "No data available for the selected criteria."

Test Graph Updates Without Full Page Refresh

Given that I apply new filters,
When the graph updates,
Then the graph reflects the new data without refreshing the entire page.

2.3.4 Save Filter Settings

As a user, I want to save my filter settings (energy types and date range), so that I can quickly reapply the same settings for future analyses.

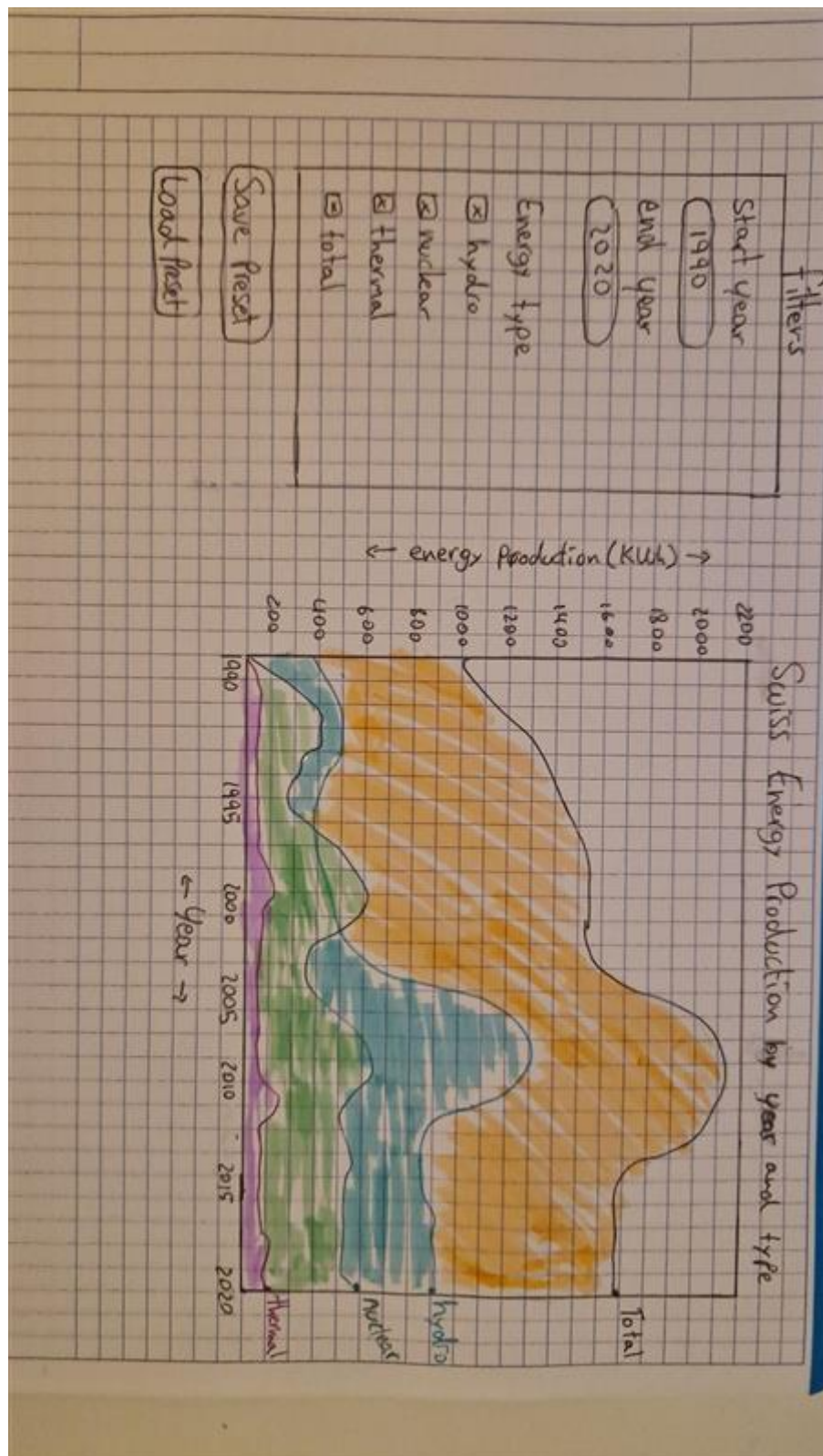
Tests d'acceptance:

Test Save Filter Stores Selected Filters

Given that I have applied filters,
When I click "Save Filter,"

	Then my current filter settings should be saved for future use.
Test Apply Saved Filter Config Loads Previous Settings	Given that I saved a filter, When I load the application later, Then I should be able to apply my saved filter config.
Test Remove Saved Filter Config	Given that I want to remove a filter config, When I click "Remove," Then the filter config should be removed from the config list.

2.3.5 Maquette de l'application



2.4 Stratégie de test

Les tests ont été effectués de manière manuelle en se basant sur les tests d'acceptance des user stories. Les tests ont été effectués sur un ordinateur personnel à la maison, ainsi que sur plusieurs postes de travail à Vennes.

3 Réalisation

3.1 Points de design spécifiques

3.1.1 Gestion des données

Pour la gestion de données de l'application. Le fichier **Excel** va d'abord être exporté en format **CSV** (manuellement). Ensuite dans l'application il y aura une lecture de ce fichier et la création de d'objets « **EnergyDataModel** ». Et pour l'affichage des objets « **EnergyDataViewModel** », seront créés selon les filtres, pour être utilisés par la vue pour la génération des graphes.

3.2 Déroulement

Le déroulement des différentes fonctionnalités c'est bien passé. Il y a eu quelques changements entre la conception du projet, et le moment de la réalisation de l'application des fonctionnalités qui ont posé quelques soucis. Comme des changements à comment l'utilisateur interagit avec l'application & la gestion des filtres, et le changement de technologie .Net pour le déroulement du projet.

Avec les diverses difficultés rencontrées certaines fonctionnalités n'ont pas pu être implémentées à cause d'un manque de temps, mais celles de base fonctionnent.

3.2.1 Fonctionnalité Filtrage

3.2.1.1 Bouton Filtrage

Le premier changement est la fonctionnalité des filtres.

Pendant la phase d'analyse fonctionnelle l'idée c'était d'avoir une zone avec tous les filtres et un bouton pour les appliquer. Le graph sera mis à jour dès que l'utilisateur applique les filtres.

Pendant la phase de codage cette fonctionnalité c'était devenu clair que l'action de devoir cliquer sur le bouton pour appliquer les filtres était moins user friendly and rendait l'application moins fluide.

Donc la fonctionnalité a été changée pour que le filtre soit automatiquement appliqué dès que l'utilisateur change un des filtres. Ce changement a été accompli dans le code avec des event listeners pour chaque composant.

3.2.1.2 Filtres invalide

Start & End Year Filter :

Un autre changement c'est comment le programme gère des filtres invalide. Pour d'analyse fonctionnelle l'idée était que l'utilisateur pouvait mettre n'importe quelle date, et que le check pour s'il y avait des données (donc la date est valide), ou pas serait faite au moment de l'application du filtre date.

Mais pour faire ce check, le programme doit parcourir toutes les données pour ensuite dire à l'utilisateur que leur requête n'est pas valide. Niveau ressources ce n'est pas optimiser.

Donc à la place on ne laisse pas la possibilité à l'utilisateur de faire une requête invalide, les années pour le filtre « **Start Year** » et « **End Year** » correspondent aux dates min et max du data set.

Energy Type Filter :

Il y a aussi eu un changement pour le filtre « **Energy Type** », à l'origine le graph devait afficher toutes les données si toutes les options du filtre n'étaient pas cochées. Comme ça l'utilisateur ne pouvait pas créer un graph vide.

Mais ce n'était pas intuitif, aucune filtre energy type cochée au lancement de l'application mais le graph contient leurs données. Ça causé juste de la confusion.

Donc le changement de mettre un check pour le filtre pour que l'utilisateur a toujours au moins un energy type coché. Et au lancement de l'application toutes energy types seront cochées pour que l'utilisateur puisse tout sélectionner, et ensuite en désélectionner si voulu.

3.2.2 Fonctionnalité Filter Preset

À la conception de l'application une des fonctionnalités principales était de pouvoir sauvegarder/télécharger un « **filter preset** » (un sort de profile, ensemble de filtres), et à un autre moment charger sont présent pour éviter le besoin de devoir les refaire.

L'idée pour l'application était d'avoir une sélection de filtres variés pour ensuite effectuer des requêtes complexes. Et pour ne pas oublier ces requêtes complexes, un système de sauvegarde.

Mais avec un nombre de filtres limitée et avec l'avancement du projet ralenti, cette fonctionnalité a perdu de l'importance et n'a pas été implémentée.

3.2.3 Technologie utilisée

Au début de projet le Framework **ASP.NET MVC** a été choisi pour son déroulement, pour les raisons suivantes :

- L'intérêt pour le Framework
- Apprendre une nouvelle tech pour renforcer les connaissances techniques
- Défis change personnel

Avec le choix fait le projet dérouler normalement, journal de travail, user stories, etc. Dans les premières semaines de projet un parti du temps a été consacré à l'apprentissage du Framework.

Avec les semaines qui avancent la faute d'avoir choisi **ASP.NET MVC** à la place de **C# Windows Forms App** semblait de plus en plus d'avoir été une erreur.

D'avoir jongler entre coder en **HTML/Javascript** et en **C#**, apprendre tout un Framework, et implémenter les notions de **programmation fonctionnelle** a considérablement ralenti le projet.

Le plus part du temps a été consacré au **Framework** et à se rendre compte que cela, avec l'application envisagée n'était pas compatible.

Un manque de connaissances, de la dette technique, et la date de fin du projet qui arrivait, a fait qu'au 5 semaine de projet (début octobre), le choix d'abandonner le projet pour recommencer à zéro a été faite.

Si ce choix n'avait pas été faite plus tout c'est à cause de l'idée du « **sunk cost fallacy** », l'idée que tant de temps et ressources ont déjà été investi, donc il faut continuer avec le projet, ce n'est pas envisageable d'abandonner.

Mais avec l'effort fourni, et sont rendu c'était la bonne décision. Donc beaucoup de retard. La version **C# Windows Form App** a été commencer durant les vacances d'octobre.

Certaines classes on put être utiliser dans la nouvelle version du projet, comme les classes « **CsvReader.cs** » et « **EnergyDataModel.cs** », avec le reste de l'application à refaire.

Le reste du projet en **Windows Form App** s'est dérouler de façon normale. Juste a implémenter les différents fonctionnalités décrits dans les user stories et s'assurer qu'ils passe les tests.

3.3 Mise en place de l'environnement de travail

GitHub repos : https://github.com/ethanschafstall/P_FUN_323

Visual Studio 2022: Visual Studio Community 2022 (64 bits) - Version 17.9.2

Librairies C# utilisé: <https://v0.lvcharts.com/>

Données :

- 5634-Zeitreihe_Elektrizitätsbilanz_Schweiz_Monatswerte.xlsx : Excel originaire de bfe.admin.ch
- 5634-Zeitreihe_Elektrizitätsbilanz_Schweiz_Monatswerte.csv : Données sous form de CSV

Fichier projet :

- EnergyDataModel.cs : Représente les données énergétiques
- FilterModel.cs : Modèle pour filtrer les années, mois et types d'énergie.
- EnergyDataViewModel.cs : Modèle pour l'affichage des moyennes de production par année.
- CsvReader.cs : Lit et traite le fichier CSV en instances EnergyDataModel.
- EnergyDataService.cs : Service qui charge et filtre les données énergétiques.
- GraphHelper.cs : Gère la génération des graphiques pour chaque type d'énergie.
- Form1.cs : Interface utilisateur principale avec graphiques et filtres.

3.4 Description des tests effectués

Filter Data by Energy Type :

Test Filter Selection Updates Graph	Given that I am on the data filtering interface, When I select one or more energy types, Then the graph updates to display only the data for the selected types.	OK 01.11
Test No Filter Displays All Energy Types	Given no energy type is selected, When I submit the filter, Then the graph displays data for all energy types.	NON* 01.11

La fonctionnalité a changé alors ce test est deprecated, donc non.

Filter Data by Date Range :

Test Date Range Filter Updates Graph	Given that I input a start and end date, When I apply the filter, Then the graph updates to show data only within the selected date range.	OK 01.11
Test Invalid Date Range Triggers Error	Given on the date filter section, When the selected end date is earlier than the start date, Then the filter button is unclickable.	OK 01.11
Test Missing Start or End Date Triggers Error	Given on the date filter section, When an incomplete date is entered, Then the filter button is unclickable.	NON* 01.11
Test No Date Selected Shows Full Data Range	Given no date is selected, When I apply the filter, Then the graph should display data for the full time period (past 30 years).	OK 01.11

La fonctionnalité a changé alors ce test est deprecated, donc non.

Display Filtered Data in Graph :

Test Graph Displays Correct Axes Labels	Given that I have applied filters for energy type and date range, When the graph is generated, Then the graph shows the correct axes for time (x-axis) and energy production (y-axis).	OK 01.11
Test Multiple Energy Types Display with Distinct Colors	Given that I have selected multiple energy types, When the graph is displayed, Then each energy type should have a distinct color or label for clarity.	OK 01.11

Test No Data for Selected Filters Shows Message	Given that no data exists for the selected filters, When the graph is displayed, Then I should see a message saying "No data available for the selected criteria."	NON* 01.11
Test Graph Updates Without Full Page Refresh	Given that I apply new filters, When the graph updates, Then the graph reflects the new data without refreshing the entire page.	OK 01.11

La fonctionnalité a changé alors ce test est deprecated, donc non.

Save Filter Settings :

Test Save Filter Stores Selected Filters	Given that I have applied filters, When I click "Save Filter," Then my current filter settings should be saved for future use.	NON* 01.11
Test Apply Saved Filter Config Loads Previous Settings	Given that I saved a filter, When I load the application later, Then I should be able to apply my saved filter config.	NON* 01.11
Test Remove Saved Filter Config	Given that I want to remove a filter config, When I click "Remove," Then the filter config should be removed from the config list.	NON* 01.11

*La fonctionnalité n'a pas encore été implémenté, donc non.

3.5 Erreurs restantes

- Dette Technique sur l'affichage du graph avec seulement d'une seule année
→ seulement des points visibles dans le graph. Par manque de connaissance du librairie **LiveCharts**
- Dette Technique nullable variables, ou dans certain parti du code la seule option était de faire le cast d'un null en **nullable double** pour créer un **EnergyDataViewModel** avec une propriété null.

4 Conclusions

Je suis déçu de ne pas avoir pu faire fonctionner l'application avec le **Framework ASP.NET MVC**, et que le choix de l'avoir utiliser ma empêcher d'implémenter certaines fonctionnalités, ou créer un interface utilisateur plus sympa.

Même si je n'ai pas réussi, j'ai beaucoup appris en utilisant le Framework, mais c'était juste trop ambitieux pour le contexte du projet, et le temps.

Dans la version Form de l'application il n'y pas eu trop de problèmes, un peu de dette technique avec les variables nullable, mais sinon c'était sympa de pouvoir utiliser les diverses notions de la programmation fonctionnelle appris durant le module.

Pour la suite du projet, ça serait bien d'implémenter la fonctionnalité des presets, de d'y ajouter les filtres par mois. Aussi faire que l'application soit plus responsive

5 Annexes

5.1 Journal de travail

[Lien du JDT](#)